

# Create Your Own Excel Add-In

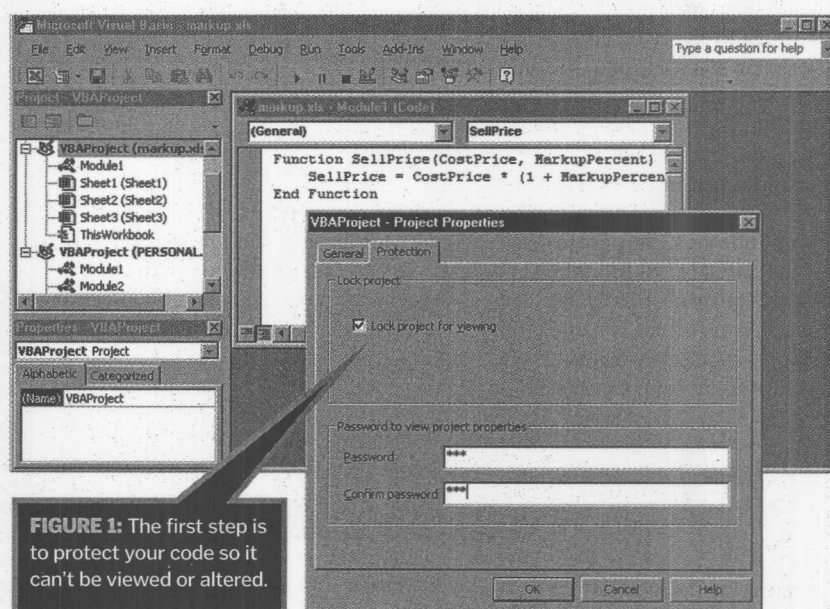
Here's how to make your custom functions easily accessible to anyone. **By Helen Bradley**

**W**riting your own Excel functions lets you perform tedious and complicated calculations more easily. In the previous issue we looked at the process of creating custom functions. Now we'll see how to make these functions accessible to others using Excel's add-in feature.

You may already be familiar with some Excel add-ins, such as the Analysis ToolPak, which contains functions that aren't built into Excel in the way that SUM and MAX are, for example. Once you've installed the Analysis ToolPak and enabled it as an add-in, however, its functions are available to any Excel worksheet. Every time Excel loads, it takes care of opening the add-in

leave the functions and macros accessible to other workbooks. There is one caveat: If you use one of an add-in's functions in a workbook you share with people who haven't installed the add-in, they'll see an error. All they have to do, though, is install the add-in, and the error will disappear.

Let's create a simple example. Open a new Excel workbook and choose **Tools | Macro | Visual Basic Editor**. Select the workbook in the Project Explorer and add a module by choosing **Insert | Module**. (If the Project Explorer isn't visible, select it from the View menu.) Enter this function, which calculates the selling price for an item with a given cost and markup percentage:



**FIGURE 1:** The first step is to protect your code so it can't be viewed or altered.

so you can forget about the mechanics of the process and concentrate on using the functions.

You can create your own add-ins from any Excel worksheet. When you do, the workbook is hidden from other users, but its functions and macros remain available. You can even lock the code so it can't be viewed or edited without the correct password, but still

```
Function SellPrice(CostPrice,
MarkupPercent) as Currency
    SellPrice = CostPrice * (1 +
MarkupPercent)
End Function
```

Now select **Close and Return to Microsoft Excel** from the **File** menu. Save the worksheet as *Markup.xls* and leave it open. To test the function you

Free  
me.

Power  
me.

Best of  
both worlds  
me.

wrote, open a new workbook and type this formula into a cell:

```
=Markup.xls!sellprice(2000,.15)
```

The result should be 2300.

To turn Markup.xls into an add-in function so that the selling-price calculator can be more

**FIGURE 2:** Enabling the add-in makes its functions available to other workbooks.

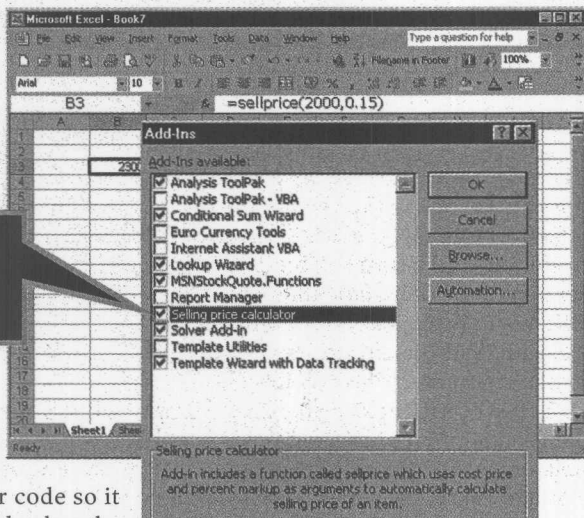
easily accessed, open the Visual Basic Editor again. (Alt-F11 gets you there quickly.)

Start by locking your code so it can't be viewed or altered unless the password is typed. (You don't have to do this, but it's a good idea to keep your code safe from unauthorized editing.) In the Project Explorer, locate and select the entry for Markup.xls. Choose **Tools | VBAProject Properties** and the **Protection** tab (Figure 1). Enable the **Lock project for viewing** check box, and to protect it, type the same password in the **Password** and **Confirm password** boxes and click **OK**. Return to Excel by choosing **File | Close and Return to Microsoft Excel**.

Although the next step is also not required, setting the **Title** and **Comments** properties of your workbook is highly recommended, because these become part of what identifies the add-in for users. Choose **File | Properties**, and then select the **Summary** tab. Type a title for your add-in in the **Title** field. This, rather than the add-in filename, is what will appear in the list in the **Add-ins** dialog, so make the title descriptive. We'll call our example **Selling-price calculator**.

Now type a sentence or two of further detail in the **Comments** field; this description will appear in the **Add-ins** dialog when the add-in is selected. Click on **OK** to exit this dialog and then choose **File | Save As**. From the **Save as** drop-down list, choose **Microsoft Excel Add-In (\*.xla)** and type a name for

the file. We will call this one *Markup.xla*. Excel automatically selects the add-in folder as the location for saving the file, so leave this as is and click



**Save**, then close the file.

To enable your new add-in, choose **Tools | Add-ins** (Figure 2). If you don't see your add-in, click on **Browse** and locate and choose your XLA file, then click on **OK**. Click on the check box to the left of your file's description (this is the text you typed for its title) and click on **OK**. You can now use any function defined in the add-in file without prefixing it with the filename. Test that by using this formula in any workbook:

```
=sellprice(2500,.15)
```

To edit a function you've made into an add-in, simply open and edit the XLA file. (If you've set the protections as we recommend, you'll need to type your password to gain access.) If you're using Excel 97, edit the original XLS workbook, then resave it as an add-in.

Distribute the add-in to others by giving them the XLA file. They can put it in their own add-ins folders, then use the **Tools | Add-ins | Browse** process to select and enable it so they can access its functions.

*Helen Bradley specializes in writing hands-on tutorials. Her columns appear regularly in a number of publications in Australia, Canada, the U.K., and the U.S. Contact her at helen@helenbradley.com.*

# AMD me.